



Damen, D., Gee, A., Mayol-Cuevas, W. W., & Calway, A. D. (2012). Egocentric Real-time Workspace Monitoring using an RGB-D Camera. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (pp. 1029-1036). 10.1109/IROS.2012.6385829

Link to published version (if available):
[10.1109/IROS.2012.6385829](https://doi.org/10.1109/IROS.2012.6385829)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>

Take down policy

Explore Bristol Research is a digital archive and the intention is that deposited content should not be removed. However, if you believe that this version of the work breaches copyright law please contact open-access@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline of the nature of the complaint

On receipt of your message the Open Access Team will immediately investigate your claim, make an initial judgement of the validity of the claim and, where appropriate, withdraw the item in question from public view.

Egocentric Real-time Workspace Monitoring using an RGB-D Camera

Dima Damen¹, Andrew Gee¹, Walterio Mayol-Cuevas^{1,2}, Andrew Calway¹

¹Department of Computer Science, University of Bristol, BS8 1UB, Bristol, UK

²Bristol Robotics Laboratory, BS16 1QY, Bristol, UK

{damen, gee, wmayol, andrew}@cs.bris.ac.uk

Abstract—We describe an integrated system for personal workspace monitoring based around an RGB-D sensor. The approach is egocentric, facilitating full flexibility, and operates in real-time, providing object detection and recognition, and 3D trajectory estimation whilst the user undertakes tasks in the workspace. A prototype on-body system developed in the context of work-flow analysis for industrial manipulation and assembly tasks is described. The system is evaluated on two tasks with multiple users, and results indicate that the method is effective, giving good accuracy performance.

I. INTRODUCTION

Effective monitoring of personal workspaces is a fundamental requirement in many robotic systems, on both autonomous and wearable platforms. Robust, accurate and real-time sensing of activities and actions within the immediate spatial vicinity of the user would allow timely and meaningful decision making. Moreover, the sensing needs to be egocentric, facilitating free movement of the user in natural environments without the need for an external sensing infrastructure. In this paper we describe the development of an approach to workspace monitoring which addresses these issues. The system aims at capturing work-flow information whilst a user carries out a hand manipulation and assembly task, such as that often encountered in bespoke industrial manufacturing.

We base our approach around an RGB-D sensor suitably mounted on the user to provide good visibility of the workspace (Fig. 1). Our aim is that the user should walk up to the workspace and monitoring is started, with minimal set up and pre-calibration of the environment. In the current application, monitoring requires the real-time detection of tools and components in the workspace, recognising them and then tracking their 3D movement as the task is undertaken. This needs to be done in real-time and with minimal mis-classification and loss of track.

There are two key components in the method. First, we use an efficient optimisation strategy to fuse the depth maps and appearance frames from the RGB-D camera to build a dense 3D reconstruction of the workspace. This then forms the basis of robust tracking and re-localisation of the sensor and the segmentation of foreground objects. Second, we use a fast object recognition method from [3] which provides reliable and simultaneous recognition of multiple previously seen objects. Importantly, recognition is based on configurations of edgelets which is well suited to the minimal-textured tools and components usually encountered in industrial manufacturing.

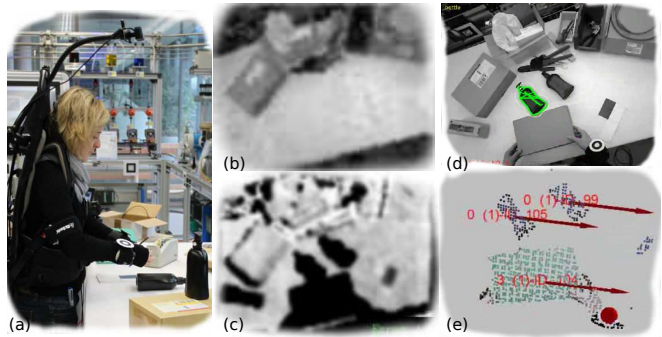


Fig. 1. Our egocentric workspace monitoring prototype (a) tracks an RGB-D sensor using dense modelling of the workspace in terms of both depth and appearance (b) and segments outliers indicating the presence of new objects (c). Known objects are then recognised (d) as well as tracked in 3D as tasks are executed (e).

The novelty in the work centres primarily on the integration of these different techniques. We are not aware of similar configurations being investigated previously and our experience indicates that it provides an effective approach to workspace monitoring of the form described. Crucially, the majority of previous methods appear to be significantly more restrictive and often require off-line processing. It is also important to note that we have tailored the techniques to match the specific requirements of our application. We have extended the dense stereo tracking method of Comport *et al* [2] to closely couple depth and appearance via a coherent optimisation strategy for fusion and tracking, and we have found that it provides greater flexibility and robustness than using just depth or appearance alone. We have also extended the object recognition method of [3] to allow the recognition of in-hand tools and components, exploiting the characteristic grips adopted by users when handling specific objects.

We have evaluated the method in two scenarios and with 5 different users completing the tasks multiple times, giving a total of 50 evaluations. The two tasks involved different tools and components and also differed in the form of actions being undertaken. In both cases, the method proved very effective. Comparison with ground-truth data indicates high performance in terms of accuracy and reliability. The experiments are detailed in Section VII. In the following sections we provide a brief review of related work and then present details of the individual components in the method. The paper concludes with a discussion on likely directions

for future work.

II. RELATED WORK

Several systems have been developed with similar aims to that described in this paper. However, in the main, these are restrictive either in terms of camera placement or in terms of the objects that can be tracked. For example, in [8] the operator is expected to sit facing a static camera and manipulate the object using the right hand. Skin colour is used to extract the face and the hand regions, and the object is detected to the right of the hand, using histogram of oriented edgelets. The work focuses on the joint recognition of the action and the object involved, showing an improved performance when conditional random fields are used to combine the action with the object recognition.

Simulated virtual reality data was used in [16] to recognise activities based on the manipulation of objects. The work emphasises that most physical activities performed by humans are mediated by objects, and distinguishes between two types of manipulation, one that does not change the state of the object and another that does. Combining multiple sensor methods like a static camera and RFID tags was used in [18] to learn objects on-line while being manipulated. The operator wore an RFID bracelet to read nearby RFID signals. Noisy RFID signals along with visual data enabled object learning without human supervision. Visual recognition of the objects used SIFT descriptors, and is thus suitable for highly-textured objects. The work jointly infers the most likely object and activity.

The recent work of Lenz *et al* [9] uses two static cameras and 8 hand sensors. The work tracks the positions of the hands in 3D using an occupancy grid. Objects are not detected, and are assumed to be in fixed pre-determined global positions. The hand reaching certain areas is thus identified as the action of picking an object. The work then uses an HMM for activity recognition and workflow analysis.

Two earlier works from 2002 [14] and 2004 [17] are similar to our objective, retrieving 3D trajectories of objects within the workspace. In [14], RGB and depth images are obtained using three static and calibrated camera and range sensors. Skin colour extracts hand regions in the image and these are compared to templates of grips using normalized correlation. Plastic objects of distinct colours are used and ICP tracking retrieves 6DoF positions and orientations of these objects within the workspace. The work is restrictive to a static sensor, and does not state the processing time. The paper's results show one worker performing a single action. In [17], a static setup of two colour cameras and two infrared cameras was used. Moving objects are then segmented into hand and object regions using skin colour, and objects are tracked in 3D. Results are limited to moving one object in an empty space.

Online learning of hand-held objects was also achieved in [13] using a static camera. Foreground segmentations, along with maximally stable extremal regions tracking, generate a collection of templates for learning. As the gathered

templates are highly correlated, incremental PCA is used for feature compression.

Using wearable cameras and egocentric views for object and activity recognition is quite recent in visual systems. The early work of Mayol-Cuevas and Murray [11] segments the hand using skin colour and represents the objects using colour histograms. The recent work of Fathi *et al* recognises hand-held objects using a head-mounted monocular camera [4]. The approach emphasises the importance of foreground segmentation to focus on manipulated objects and uses skin colour to segment hand regions. The foreground regions are estimated by fitting a fundamental matrix to dense optical flow vectors. The method is though far from being used in real-time, as the used techniques like super-pixel segmentation, SIFT descriptors and graph cuts are not suitable for a real-time performance.

Similarly, the work of Sun *et al* uses a wearable gaze-positioned camera [15]. Skin colour is used to segment hands, and edges combined with CAD models are used to localise the objects. Three-dimensional models of the hand are used to identify the grip in 27 degrees of freedom, which is then combined with object positions and identities to recognise the activity. The system was tested on two objects: a cup and a milk box, and provides trajectories of these objects using offline processing.

III. OVERVIEW OF THE METHOD

The method consists of 5 main components as illustrated in Fig. 2. These are outlined below.

Background mapping

Prior to task execution, a dense 3D map of the workspace is constructed by fusing data from the RGB-D camera (Asus Xtion Pro Live). We adopt an approach with a framework akin to the depth fusion method recently described in [12], although we incorporate appearance information and use an alternative optimisation strategy.

Tracking and relocalisation

The dense 3D model of the workspace then allows alignment and hence tracking of the 6D pose of the sensor. Again this is achieved by combining both depth and appearance within the same optimisation strategy as that used for map building. In parallel we capture feature descriptors (SURF) [1] for salient points in the map which allows fast re-localisation in the event of tracking loss.

Foreground segmentation

An important by-product of the optimisation process in tracking is the automatic identification of outliers inconsistent with the depth and appearance of the workspace map. These

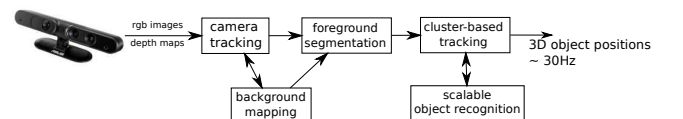


Fig. 2. Proposed System's Workflow

correspond to the presence of new objects in the scene and hence allow foreground segmentation. This is particularly important for real-time operation since it enables the focusing of image processing on relevant regions.

Cluster-based tracking

Foreground regions of significant and consistent connected size are then tracked as they move within the workspace providing the required 3D trajectory information. Due to the quality of the foreground segmentations, we have found that a simple and fast frame-to-frame cluster association process is sufficient to give good performance.

Scalable object recognition

For each segmented foreground cluster we apply the scalable object recognition algorithm described in [3] which enables real-time learning and recognition of individual components and hand-held tools. We have adapted this to allow online learning within the workspace prior to task execution and show that including hand held examples in the data set significantly improves recognition during task execution.

The above method provides real-time 3D tracking, segmentation and recognition of individual components and tools as they are manipulated within the workspace. Further details of each component are given in the following sections.

IV. MAPPING AND CAMERA TRACKING

The tasks of mapping, camera tracking and foreground segmentation are performed using a dense RGB-D Simultaneous Localisation and Mapping (SLAM) technique. The method is an extension of the quadrifocal visual odometry system proposed by Comport et al. [2], which minimises intensity error to perform dense spatial matching between pairs of stereo images. The recent commercial availability of RGB-D cameras has enabled the modification of this approach to perform combined minimization of intensity and depth information. This provides robust tracking and segmentation in both textured and untextured environments and avoids the computationally expensive dense stereo correspondence search. Furthermore, a textured occupancy grid representation of the scene is generated by fusing depth and intensity images over multiple frames. This enables tracking to be performed relative to reference images extracted from the fused model, which reduces tracking drift compared to frame-to-frame tracking methods [12]. Figure 3 shows an overview of the tracking and mapping components.

A. Tracking

Camera tracking estimates the global camera pose, $\mathbf{T}^{wc} \in \mathbb{SE}_3$, at each new frame using the information contained in the current image and the stored map. We use notation \mathbf{T}^{fo} to represent the pose of object o w.r.t. coordinate frame f , where f and o take values w , c and r for the world frame, current camera frame and reference camera frame respectively. The current RGB-D image pair, $\mathbf{I} = \{\mathbf{I}_D, \mathbf{I}_I\}$, contains a depth image, \mathbf{I}_D , which has been registered to the same viewpoint as the intensity image, \mathbf{I}_I . A similar image pair, $\mathbf{I}' = \{\mathbf{I}'_D, \mathbf{I}'_I\}$, can be generated from the textured

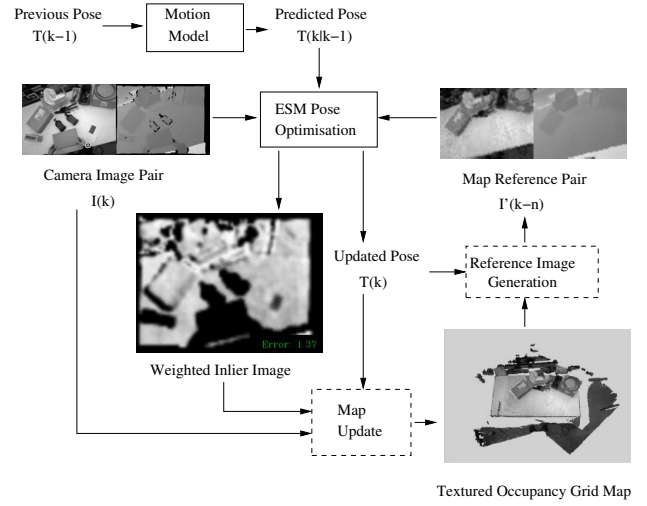


Fig. 3. The mapping and camera tracking system. Mapping components in dashed boxes are run in a background thread to enable tracking to run at 30Hz. Individual components are described in Section IV.

occupancy map by ray casting from a nearby, previously estimated, camera pose, \mathbf{T}^{wr} , as described in Section IV-B. From now on, \mathbf{I} will be referred to as the *current* view pair and \mathbf{I}' as the *reference* view pair.

Suppose that we have an estimate, $\tilde{\mathbf{T}}^{rc} \in \mathbb{SE}_3$, of the rigid-body transformation between the current and reference view, initialised by applying a decaying constant velocity motion model to the previous estimated camera pose. Then the initial estimate of camera pose for the current view is $\tilde{\mathbf{T}}^{wc} = \tilde{\mathbf{T}}^{rc} \mathbf{T}^{wr}$. Using this estimated pose, it is possible to warp the reference view images to the current view as follows

$$\mathbf{p} = \mathbf{K} \tilde{\mathbf{T}}^{rc} \mathbf{I}'_D(\mathbf{p}') \mathbf{K}'^{-1} \mathbf{p}' = \mathbf{K} w(\mathbf{p}', \tilde{\mathbf{T}}^{rc}) \quad (1)$$

where \mathbf{p} and \mathbf{p}' are pixel coordinates in the current view and reference view respectively, $\mathbf{I}'_D(\mathbf{p}')$ is the depth of pixel \mathbf{p}' in the reference depth image, and \mathbf{K} and \mathbf{K}' are intrinsic camera matrices for the current view and reference view.

The optimised pose, \mathbf{T}^{rc} , is found by minimizing a non-linear cost function formulated as a weighted combination of the intensity and depth differences

$$C(\mathbf{x}) = \sum_{\mathbf{p}' \in \mathbf{I}'} \rho(\alpha r_D(\mathbf{p}', \mathbf{x}) + (1 - \alpha) r_I(\mathbf{p}', \mathbf{x})), \quad (2)$$

$$r_D(\mathbf{p}', \mathbf{x}) = \mathbf{I}_D(\mathbf{K} w(\mathbf{p}', \tilde{\mathbf{T}}^{rc} \mathbf{T}(\mathbf{x}))) - (w(\mathbf{p}', \tilde{\mathbf{T}}^{rc} \mathbf{T}(\mathbf{x})))_z, \quad (3)$$

$$r_I(\mathbf{p}', \mathbf{x}) = \mathbf{I}_I(\mathbf{K} w(\mathbf{p}', \tilde{\mathbf{T}}^{rc} \mathbf{T}(\mathbf{x}))) - \mathbf{I}'_I(\mathbf{p}') - \tilde{I}, \quad (4)$$

where $\mathbf{T}(\mathbf{x})$ is an incremental transformation generated from the exponential map of the incremental transform parameters, $\mathbf{x} \in \mathbb{R}^6$, in the \mathbb{SE}_3 tangent space, $\rho(r)$ indicates the use of Tukeys robust M-estimator function [2], α is a weighting parameter, $(\dots)_z$ indicates selection of the z component of the vector, and \tilde{I} is an estimate of median intensity difference that models global illumination changes between the current and reference views and is included as an additional parameter to be optimised [6].

Iterative minimization of the cost function (2) is performed using the efficient second-order approximation (ESM) [10]. For reliable real-time operation at 30Hz, we run five iterations each frame on 80×60 pixel downsampled images and set $\alpha = 0.95$. Note that an extra benefit of this approach is that we can use the same map to track monocular cameras by simply adjusting α ($\alpha = 1.0$ for depth-only, $\alpha = 0.0$ for monocular intensity). Additionally, the outlier rejection weights from the Tukey M-estimator are used to generate a *weighted inlier image* that enables simple segmentation of foreground objects from the map, as described in Section V.

B. Mapping

Successfully tracked frames are fused into a textured occupancy grid map representation in order to build a representation of the static environment. This is implemented efficiently using an extended version of the octree representation and probabilistic update framework provided by the OctoMap library [19]. Modifications include a customised sensor model for RGB-D depth data, the ability to store estimated intensity values alongside the occupancy probabilities for each voxel, and an optimised reference image generation routine based on the information in the weighted inlier image. In order to support real-time operation, the map update and reference image generation are run in a separate thread to the tracking.

Since the Asus Xtion Pro Live RGB-D camera uses a stereo approach to estimate depth, the depth error can be modelled as $\sigma_d = kd^2$ [7], where σ_d is the depth standard deviation at distance d from the camera and k is a constant of proportionality that depends on the focal length and baseline of the IR camera and projector. The sensor model defines the probability of occupancy for a voxel as

$$P_v = \begin{cases} P_{min} & \text{if } d - v_d > 3\sigma_d \\ 0.5 & \text{if } d - v_d < -3\sigma_d \\ \frac{1}{2} \left(1 + \int_{v_d - \frac{s}{2}}^{v_d + \frac{s}{2}} \mathcal{N}(d, \sigma_d^2) \right) & \text{otherwise} \end{cases} \quad (5)$$

where v_d is the depth of the voxel from the camera, s is the size of the voxel, and $\mathcal{N}(d, \sigma_d)$ is a normal distribution representing the measured depth d at the backprojected location of the voxel in the depth image. The value of P_{min} is set to 0.4 (as in [19]) and a voxel is said to be occupied if $P_v > 0.51$. Additionally, instead of updating the probabilities of all voxels in the viewing frustum of the camera, only the voxels that lie in the range $[d - (3\sigma_d + 3s), d + (3\sigma_d + 3s)]$ of the measured image depths are updated. This significantly increases the speed of scan insertions at the expense of maintaining a fully updated set of freespace cells.

Intensity values are stored for each occupied voxel in the map. Since intensity is subject to large global and local variations, the stored value in each voxel is simply overwritten by each new image that is fused into the map.

The map also needs to be able to generate reference image pairs for use by the tracker. Given a camera pose, \mathbf{T}^{wc} , the image pair, \mathbf{I}' , is generated from the textured occupancy map by raycasting from each pixel. Along each ray, the closest surface is found by searching for the first maxima

in the occupancy probability that exceeds the occupancy threshold. During normal tracking operation, this search can be accelerated by using the weighted inlier image from the last tracked frame to constrain the searched region of the ray. Areas of the weighted inlier image with high inlier scores have reliable depth estimates that can be used as a prior for the maximum occupancy search.

C. Relocalisation

In order to support relocalisation when the tracking fails, the system stores a sparse set of keyframes based on simple distance and angle constraints. SURF features [1] are extracted from the intensity image in each keyframe and their global 3D position found from the depth image and estimated camera pose. When tracking is lost (indicated by a large RMS error score after pose optimisation) the system attempts to relocalise by matching SURF features extracted from the current image pair to the full set of SURF features extracted from keyframes. The set of matches are used to estimate 3D pose using RANSAC and three-point pose estimation [5].

V. OBJECT SEGMENTATION

After the camera's position is tracked within the generated map, the weighted appearance and depth differences are used to segment foreground pixels. Figure 4 shows the weighted image, the corresponding input point cloud, and the resultant foreground segmentations. Edge discontinuities produce noisy segmentations, which are cleared using image-based erosion. The foreground pixels are converted into a 3D point cloud, and are passed to the cluster-based tracker (Section VI).



Fig. 4. The weighted inlier map (darker for outliers or missing depth values) is capable of segmenting the blue label that is at the same depth as the background but of different appearance.

During segmentation, the weighted combination of appearance and depth differences is essential for segmenting objects like the box for example, where the majority of the object's points (i.e. the base of the box) are at very similar

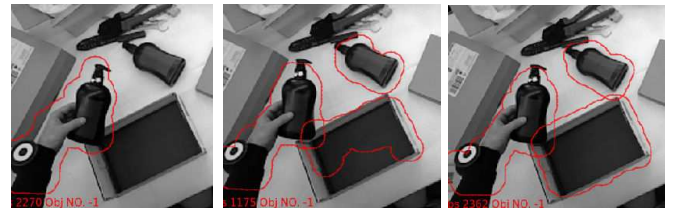


Fig. 5. As the weighting of appearance difference increases, the box is correctly segmented from the background.

depth as the background. Weighting depth and appearance differences assists in segmenting the object in full. In Fig. 5, the foreground segmentation masks are shown using different weightings of appearance and depth differences.

VI. OBJECT TRACKING AND RECOGNITION

For each frame, the foreground segmentation is clustered into connected components based on 3D spatial proximity of the points in the foreground point cloud. Clusters smaller than a set threshold (40 points in the experiments) are ignored. Clusters are then assigned to trajectories maintained from the previous frame based on spatial proximity and size similarity. New trajectories are created for unassigned clusters (larger than a set threshold of 100 points). Trajectories without any assigned clusters for a consecutive number of frames (5 in the experiments) are removed. These thresholds are kept fixed for both scenarios and all objects.

The tracker operates at 30fps, and uses object recognition as a service - i.e. selected foreground segmentations are tested for identity recognition. For each new trajectory, the clustered points are projected to 2D and are dilated to produce an image-based mask. The masks are passed for recognition, and results are fed back to the tracker with the identities of any known objects. The tracker feeds further masks from the same trajectory until the number of consistent recognised identities exceeds a certain count and ratio. In the experiments below, at least 4 consistent identities with a recognition ratio of 80% are required before the identity is trusted and the trajectory no longer feeds its masks for recognition.

The tracking module outputs the positions of all objects (known or otherwise) in 3D at 30fps, maintaining a trajectory ID until the object is mostly occluded, merges with another object or is out of view. Each trajectory maintains a single identity. While this is acceptable for the majority of the cases, situations exist where multiple known objects are clustered together. Currently, the tracker only maintains one identity and the other objects are not reported until seen separately again.

Object recognition is based on the fast and scalable real-time detection of multiple known objects from [3]. The method is shape-based and uses constellations of edgelets to describe the shape of the object. The sparse nature of constellations facilitates recognition in the presence of occlusion. Each constellation is described by an affine-invariant descriptor, defined in terms of the relative orientations and relative positions of the constituent edgelets. A key feature of the method is using *fixed constellation paths* for both training and testing. Each path defines relative directions connecting the constellation's edgelets, and these relative directions are used in extracting edgelet constellations during both training and testing, thus constraining the number of possible constellations and enabling frame-rate performance. Constellations over these fixed paths are extracted exhaustively from training views of the object. These views are evenly-sampled images from sequences of the operator holding the object to be learnt, and placing it on the workbench. Descriptors

of these constellations are calculated and arranged into a hierarchical hash table. During test time, sample constellations over the same fixed paths are extracted from the test image, and their descriptors are compared to the hash table. When matched, an affine transformation is estimated and the remaining of the shape's edgelets are used to confirm the recognition.

Experiments in [3] prove that the approach could tolerate up to 70% clutter. To handle higher percentage of clutter or smaller objects in the image, foreground segmentation is used in the system described here. All edges within the segmented masks are considered for possible edgelet constellations. Even when the masking is not perfect, due to holes in the depth map, or additional clutter, the method is capable of recognising the known objects. This is because the method is essentially designed to handle clutter and noisy edge maps.

We adopt the approach of jointly recognising the hand and the hand-held object or tool. The effect of this joint recognition depends on the object's size. For objects that extend beyond the gripping point, the object's shape remains distinctive enough when held. For other objects, often small like holding a pen, and sometimes large like holding an electric screwdriver, the grip occludes the majority of the object's shape and is influential in our ability to recognise the object being held. Due to the scalability of the method, all sampled views are inserted into the hash table, and there is no need to cluster similar views or remove duplicate ones.

The recognition module was run as a service and at a minimum rate of 10fps. Each frame is processed for a maximum of 100ms. If testing terminates before the maximum time is reached, the recognition moves to the most recent request in the service's buffer. If the maximum time is otherwise reached, the recognition abandons the current frame and proceeds to the latest request in the buffer as well. By running as a service to the tracker, the recognition focuses on new or unknown objects while the tracker maintains the identities of unoccluded objects.

VII. EXPERIMENTS AND RESULTS

Two activities were tested using different operators. The first is referred to as the 'Nails and Screws (N&S) task' and

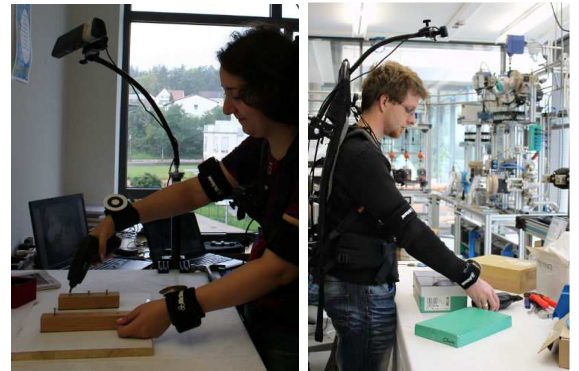


Fig. 6. The mounting of RGB-D camera during the 'Nails and Screws' task (left) and on the backpack during the 'Packaging Bottles' task (right).

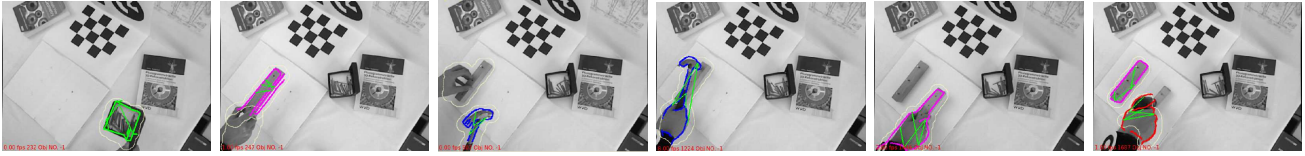


Fig. 7. Frames showing the Nails and Screws task with some recognised hand-held tools.

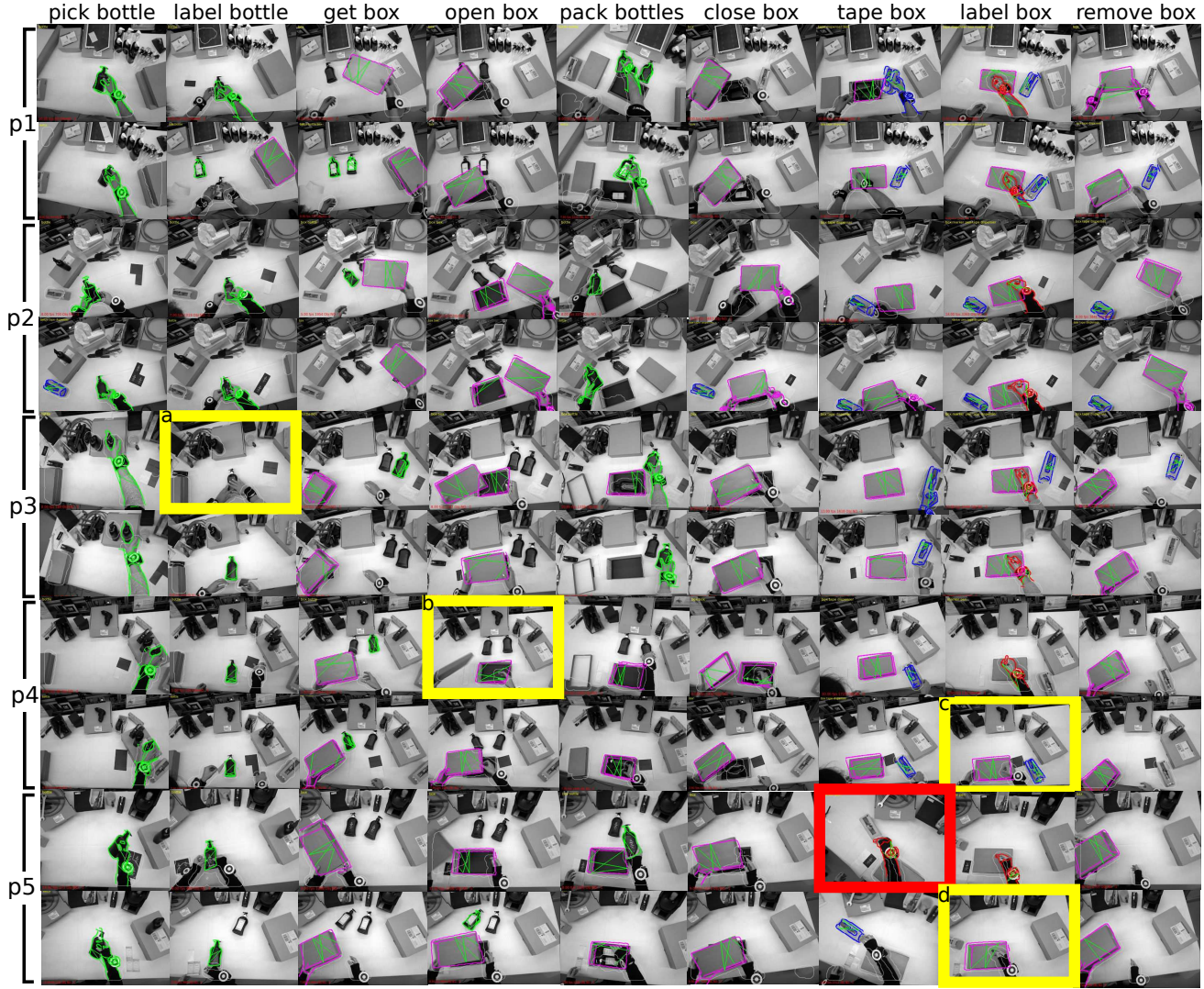


Fig. 8. Frames showing packaging bottles task for 10 sequences from 5 people (p1-p5) across the task's primitive events with some recognised tools and objects. The figure shows four false negatives in yellow due to occlusion (a), unseen pose of the lid (b), and merged clusters (c,d). A false positive in red hallucinates the marker pen as the worker approaches the tape dispenser.

the second as the 'Packaging Bottles (PB) task'. In the N&S task, the operator attaches two batons to a piece of wood. The first baton is attached using three nails retrieved from a box. A hammer is used to fix the nails to the baton. The other baton is attached using screws and an electric screw driver. The task thus involves four different tools and objects to be recognised: the box, the hammer, the screw driver and the baton. In this task two instances of the baton are used and one from each of the other classes. This task was tested using a static RGB-D camera mounted at the side of the workplace. Figure 7 shows a sequence of frames from the task along

with some recognition results. Note that the markers visible in the scene were only used for ground truth, and are not used by the system. 25 sequences were recorded using 5 operators, each performing the task 5 times.

The second task of packaging bottles (PB) (Fig. 8) also contains four object classes: a packaging box, soap bottles, a tape dispenser, and a marker pen. During the activity, two bottles are labelled by first retrieving the bottle, then attaching a label to the bottle. After labelling both bottles, the box is brought in and its lid is opened. Both bottles are placed into the box before closing the lid. The tape dispenser

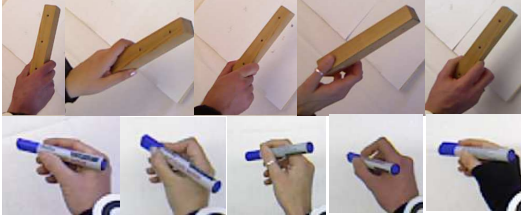


Fig. 9. Different grips for the same tool from five different operators

is then used to retrieve two tape segments that help close the box firmly. Finally, the box is labelled with the delivery address and placed aside. The egocentric prototype is used for recording the sequences. Due to the requirement of the used sensor, it has to be mounted at least 60cm from the area where depth information is required. The RGB-D camera was thus attached to a backpack using an adjustable stand (Fig. 6). 25 sequences were also recorded using five operators (see accompanying video).

During the data capture, the map was built in real-time as the operator approaches the working space and surveys the area by rotating the body to the right and left slowly. Then, each operator was asked to manipulate each of the objects used in the task in turn. The system sampled these manipulation sequences at 10Hz and the views were used to train the individualised codebook. During both mapping and learning, the operator was given verbal instructions, first to rotate the body and then to manipulate the objects in turn. Figure 9 shows different ways in which the operators handled the tools in the learning sequences.

To compare the performance of the system for the different operators, one sequence from each operator was randomly chosen and manually ground truthed with bounding boxes reflecting the different objects present in each frame. In comparing to the ground truth, the 3D positions were projected back to 2D given the camera’s position and viewpoint. A

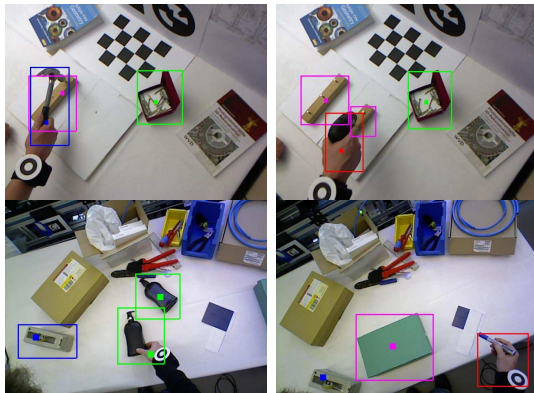


Fig. 10. Four frames showing ground truth bounding boxes and corresponding output of the method projected onto 2D images for the N&S task (top) and PB (bottom). Matching colours indicate correct identities. In the top right figure, one baton is occluded and was not tracked by our method resulting in a false negative. In the bottom right, the tape dispenser was not ground truthed correctly (false-positive), and the pen not recognised resulting in a false negative.

	# of frames	# of GT objs	recall		precision	
			μ	σ	μ	σ
person1	864	2513	67.25	0.02	94.56	0.03
person2	588	1711	72.09	0.03	90.44	0.02
person3	479	1298	68.64	0.02	87.88	0.02
person4	2181	6533	71.03	0.06	88.27	0.05
person5	2778	7621	61.81	0.02	93.31	0.02
avg			68.16		90.89	

TABLE I

FOR THE N&S TASK, AVG. PRECISION AND RECALL FROM 10 RUNS.

true positive occurs when the position of the object falls within the ground truth bounding box. Figure 10 shows four examples of comparing to the ground truth. Tables I and II show quantitative results for the ground truthed sequences. People performed the task at the speed they chose, resulting in different sequence lengths. As the tables show, the speed at which the task is performed does not affect the method’s accuracy. Results also show that both the static and egocentric camera positions produce similar results, though the egocentric position gives a better coverage of the workspace and the required flexibility.

For the above results, the manipulation sequences from the operator are first used for learning the descriptors. We test how distinctive the descriptors are by learning from the manipulation sequence of one operator then testing on all operators, and present results in a confusion matrix (Fig. 11). For the second operator in N&S task for example, the accuracy drops from 67.0% to a maximum of 29.4% when a different individual’s manipulation sequences were used for learning. For the PB task, the maximum drop was from 67.5% to 39.5% for the third operator. Average recall of 68% (N&S) and 69% (PB) are mostly due to the object being mostly occluded during the task.

We also test the average accuracy for recognising different objects within the tasks (Fig. 12). In the figure, the box in N&S task achieves the highest accuracy (94%). This is because the box was placed at the far end of the workspace and was rarely occluded. For the PB task, the box also achieved the highest accuracy as it is too big to be fully occluded. On the other end, the marker pen achieves low accuracy (40%). This is not because of the difficulty in recognising the shape, but due to the limitation of the cluster-based tracker. The cluster of the hand holding the pen is merged with the box’s cluster while writing. It was mentioned in Section VI that the

	# of frames	# of GT objs	recall		precision	
			μ	σ	μ	σ
person1	4635	12793	57.72	0.03	88.67	0.02
person2	2768	5631	79.53	0.04	77.36	0.04
person3	3807	8730	66.81	0.03	90.70	0.02
person4	3128	5523	75.99	0.05	77.15	0.05
person5	3259	6613	66.50	0.01	66.82	0.01
avg			69.31		80.14	

TABLE II

FOR THE PB TASK, AVG. PRECISION AND RECALL FROM 10 RUNS.

		Training from Person				
		1	2	3	4	5
N&S	1	64.2	56.9	58.3	46.3	59.1
	2	26.5	67.0	28.1	26.8	29.4
	3	21.3	43.4	64.9	14.7	27.2
	4	71.3	33.9	63.8	65.0	61.7
	5	34.3	40.6	46.1	31.2	59.2

		Training from Person				
		1	2	3	4	5
PB	1	54.4	44.8	44.7	40.5	41.2
	2	57.5	64.5	53.7	61.0	49.6
	3	68.5	59.3	67.5	39.5	65.8
	4	55.8	59.1	56.7	62.2	53.2
	5	45.4	52.3	39.7	43.0	51.1

Fig. 11. Confusion matrix that shows the accuracy when objects are learnt from a different operator for the N&S (left) and PB (right) tasks.

tracker currently maintains a single identity for each cluster. The pen's identity is thus often ignored by the tracker, though it achieves good recognition results.

VIII. CONCLUSION

The paper presents a system for real-time object recognition and 3D tracking during manipulation tasks. For an egocentric RGB-D camera, a map of the workspace is built in real-time and fuses depth and appearance information. The sensor can then be tracked, and outliers in the map define foreground segmentations. By clustering 3D foreground points, cluster-based tracking provides the 3D positions of clusters within the workspace. Shape-based object recognition allows identifying previously seen hand-held tools and objects. The recognition is based on constellations of edgelets described using an affine-invariant descriptor. By limiting the considered constellations to those that follow previously specified fixed paths, the approach is both fast and scalable. Recognition combines the hand-grip with the object enabling recognition of small or mostly-occluded object during manipulation. The work studies the effect of individualised grips, and shows that individualised learning often improves the performance due to the peculiarity of the grips of different operators even for the same object. This highlights the importance of online learning as part of the system. The system enables online learning of hand-held tools by sampling views from manipulation sequences.

The approach was tested on two tasks using five operators. The first task is an assembly task involving batons, hammer and electric screwdriver, and the second is a labelling and

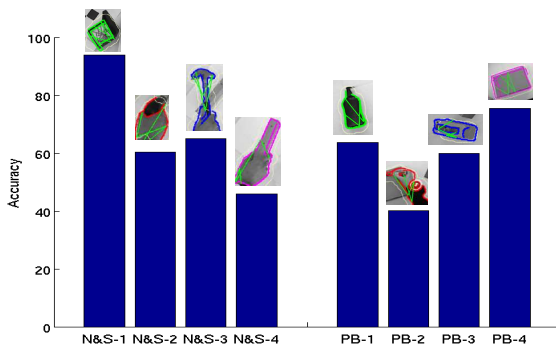


Fig. 12. Average accuracy levels for the 8 objects used in both tasks.

packaging task. Results show the ability of the method to track objects during manipulation.

Future improvements of the system would focus on cluster-based tracking and pose-estimation. Currently, the approach expects a single known object within each cluster, which is not always the case. Providing 6 DoF of the manipulated tools is a required feature for detailed workflow monitoring.

Acknowledgement This work was supported by the EU-FP7 (Project COGNITO, ICT-248290).

REFERENCES

- [1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 3, no. 110, 2008.
- [2] A. I. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," *International Journal of Robotics Research (IJRR)*, 2010.
- [3] D. Damen, P. Bunnun, A. Calway, and W. Mayol-Cuevas, "Real-time learning and detection of 3D texture-less objects: A scalable approach," in *British Machine Vision Conference (BMVC)*, 2012.
- [4] A. Fathi, X. Ren, and J. Rehg, "Learning to recognise objects in egocentric activities," in *Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [5] M. A. Fischler and R. C. Bolles, "RANDOM Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, 1981.
- [6] T. Gonçalves and A. I. Comport, "Real-time direct tracking of color images in the presence of illumination variation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [7] K. Khoshelham and S. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, 2012.
- [8] H. Kjellstrom, J. Romero, D. Martinez, and D. Kragic, "Simultaneous visual recognition of manipulation actions and manipulated objects," in *European Conference on Computer Vision (ECCV)*, 2008.
- [9] C. Lenz, A. Sotzek, T. Roder, M. Huber, and S. Glasauer, "Human workflow analysis using 3D occupancy grid hand tracking in a human-robot collaboration scenario," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [10] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *International Conference on Robotics and Automation (ICRA)*, 2004.
- [11] W. Mayol-Cuevas and D. Murray, "Wearable hand activity recognition for event summarization," in *International Symposium on Wearable Computers*, 2005.
- [12] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.
- [13] P. Roth, M. Donoser, and H. Bischof, "On-line learning of unknown hand held objects via tracking," in *Second International Cognitive Vision Workshop*, 2006.
- [14] Y. Sato, K. Bernardin, H. Kimura, and K. Ikeuchi, "Task analysis based on observing hands and objects by vision," in *International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [15] L. Sun, U. Klank, and M. Beetz, "EyeWatchMe - 3D hand and object tracking for inside out activity analysis," in *Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2009.
- [16] D. Surie, F. Lagriffoul, T. Pederson, and D. Sjolie, "Activity recognition based on intra and extra manipulation of everyday objects," in *International Symposium on Ubiquitous Computing Systems*, 2007.
- [17] Y. Tsubuku, Y. Nakamura, and Y. Ohta, "Object tracking and object change detection in desktop manipulation for video-based interactive manuals," in *Advances in Multimedia Information Processing*, 2004.
- [18] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg, "A scalable approach to activity recognition based on object use," in *International Conference on Computer Vision (ICCV)*, 2007.
- [19] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.